Computer Networks Notes

What is Network?

All computer connected together that is the network

Types of Network?

1. **LAN (Local Area Network):** small houses or office computers are connected with each other through Ethernet cables, wifi this known as local area network.
2. **MAN (Metropolitan Area Network):** Computers are connected across the city.
3. **WAN (Wide Area Network):** Computers are connected across over the country**.** Using optical fibre cables. example: The internet.
   SONET: synchronous optical networking it carries the data using optical fibre cables hence it cover larger distances.
   Frame Relay: it is basically a way to connect your local area network with a wider area network.

What is Internet?

Internet is basically a collection of this network, how every computer is connected to each other on a global scale is known as internet.

What is protocol?

Some rules is set by people how a particular thing is sent or working on the internet, is known as protocol. **The Internet Society** is set all of this rules.

Types of protocol?

1. **TCP (Transmission control protocol):** It ensures that data is sent completely and order format. It establish connection before sending.
2. **UDP (User Datagram protocol):** It is faster but does not guarantees that data is send completely and also in order, also it does not make any connection before sending data.
3. **HTTP (Hypertext transfer protocol):** It is used to transfer web pages on the internet.
4. **IP (Internet protocol):** Helps to identify and route data to correct device.

5. **DNS (Domain name system):** it is like a phonebook, it converts website name that you enter into actual IP address.

What is WWW?

WWW stands for **World Wide Web,** it is the fair document and resources identified by **URL** which may be interlinked by hyperlinks and that links are accessible over the internet. It is the collection of web pages and everything.

What is Client & Server?

Suppose you enter some URL on your computer (www.google.com) and hit the enter button then the request is sent to google's server using a GET Method, then server perform some basic operations and then sent you a response based on your request. So in this process your computer is a Client that sends a request and receive a response, and googles server is a actual server that receive a request from your computer and then give you a response. Your computer may also works on both server & client when you are working with localhost.

What is IP Address?

An IP (Internet protocol) address is a unique number assigned to every device connected to a network (like a computer, phone, or website). It acts like a phonebook suppose you call your Dad it convers name to some digits same if you type something it converts this into some number ex. 192.168.1.1 and search it on the internet.

What is port Number?

A port number is like a door on your device that helps it communicate with correct application or service, it is like your apartment room number helps deliver the right data to the right app. ex: 443,267 etc.

What is ISP?

ISP Stands for the internet service provider, it is basically a service that gives you a internet connection on your laptop or a computer via broadband, fiber, mobile data. For example, Airtel, Vi, Jio and etc. In India tear-1 service provider is TATA and tier-2 service provider is Airtel and all.

What is internet speed?

Internet speed means how fast you can get your things from the internet. it has two types 1. Upload, 2.Download.

Upload means how much time it takes to send something from one computer to another.

Download means how much time it takes to received something from the internet

Various types of Internet speed is

KBPS (Kilo bit per second): it means 1000bits/s here you can share 1000 bit per second (Slowest)

MBPS (Mega bit per second): 10 to the power 6 bits/s (1000000 bits/s)

GBPS (Giga bit per second): 10 to the power 9 bits/s

How Countries connected to each other globally?

Countries are connected to each other through a submarine cable inside the oceans, there is no any connection establish in the sky but it is inside the ocean (physically connection). This cable is faster than satellites so that it is in use.

This is the global entity used by every country to connect with each other this global entity gives the small entity to the ISP and then ISP provide us things.

What is Modem?

Modem is used to convert digital signals into analog signals and vice versa. For example that you have a digital data into your computer modem converts this data into electrical signals so that you can transfer it some telephone lines etc.

What is Router?

Router routes the data packets based on their IP Addresses., it is a network that connects different networks together and forwards data from one to another, it is like a traffic controller – it decides the best path for sending data from your device to destination on the internet.

What is Topologies?

Topology means how devices like computers, switches, routers are interconnected and communicate to a network. It is an arrangement of different elements (nodes, links, devices) in a computer network.

Types of Topologies:

1. **Bus topology:** All devise shares a single communication line (backbone), it is easy to install but high chances to colloide.
2. **Star topology: All** devices are connected to a central hub or switch, it is easy to manage, but if hub failures then whole network brings down.
3. **Ring topology:** Each device is connected to two others, forming a circle, but if one device fails the whole ring can break.
4. **Mesh topology:** Every device is connected to every device, data always finds a path but very expensive and complex setup.
5. **Tree topology:** Combination of star and bus topologies, it is more scalable and easy to manage but it root node fails then entire tree fails.
6. **Hybrid topology:** Mix of two or more topologies, flexible and adaptable but complex to design

Structure of the network

Network is a very complex thing to understand so it is important to break everything for easily understand like how things are working at each level.

For this OSI model is main model to learn internet. However it Is not for a practical use.

OSI Model (Open System Interconnection model)

1. **Application layer:** it is implemented in the softwave when user interact with some applications, sends messages, emails this all handles by the applications.you send the data to the next layer(Presentation layer). it does not refer to actual application like chrome or outllok but rather to network services those applications use.(e.g. HTTP, FTP,SMTP).
   Application layer Responsibilities: Provides a platform for user application to interact with the network, also helps in data translation
   Application layer protocols:
   1. HTTP/HTTPS: Used for web browsing and transferring web content.

2. FTP: Used for transferring file between client and server
3. SMTP: Sends mail from client to server.
4. DNS: Resolves domain name to IP address.
5. TELNET: Remote login to another system.
6. SNMP: Manages and monitors network devices.

How it works (Flow of data):

1. **User Interaction**: The user opens a web browser and enters a website URL.
2. **Application Layer (HTTP)**: The browser (HTTP client) sends an HTTP request to the server.
3. **Presentation Layer**: Data is formatted (e.g., HTML, encryption if HTTPS).
4. **Session Layer**: Manages the session between client and server.
5. **Remaining OSI Layers**: Data flows down to the lower layers (Transport → Network → Data Link → Physical).
6. **Server Response**: The same stack is used in reverse to send a response back to the client.

2. **Presentation Layer:** Presentation layer takes all the data sends by the application layer and convert it into machine level code or binary code. This layer is known for translation, encryption, compression and formatting before the data is transmitted it goes under encoding it encrypts the data and also compressed it. Then data is sent to **session layer.** It acts as translator between the application layer (Layer 7) and transport/network layer (Layer4/3).Its often called the "Syntax Layer" because it formats and transforms data syntax between systems.

Main Functions of the Presentation Layer:

1. **Translation**: Converts data between application formats and network formats (e.g. from EDCDIC to ASCII).
2. **Data** Serialization: Converts complex data structures into byte streams and vice versa.
3. **Encryption & Decryption:** Ensures data security by encrypting before sending and decrypting upon receiving.
4. **Compression & Decompression:** Reduces data size to save bandwidth and improve speed.

5. **Character Set Conversion:** Handles different encoding formats like ASCII, Unicode, UTF-8
6. **Formatting**: Converts images, audio, or documents into standard formats (JPEG, MP4, PDF, etc.)

3. **Session Layer:** session layer handles all the session established, managing and termination before sending or receiving the data between two devices (host) in a network. It handles all the authentication and authorization part.

   Function of a Session Layer:

   1. **Session Establishment**: Opens a session between two systems before data transfer.
   2. **Session Maintenance:** Maintain and keeps track of the session during communication.
   3. **Session Termination:** Gracefully closes sessions once communication ends.
   4. **Synchronization:** Inserts checkpoints into data streams to allow resuming after interruption.
   5. **Dialog Control:** Manages who can send and when (full-duplex or half-duplex).
   6. **Authentication:** Optional login/authentication before sessions begins (used secure sessions).

4. **Transport Layer:** The **Transport Layer (Layer 4)** is responsible for **end-to-end communication**, ensuring **reliable or unreliable delivery** of data between devices across a network. It breaks large data into segments and ensures **data integrity**, **ordering**, and **flow control** between sender and receiver applications.

   Key Functions of the Transport Layer

   1. **Segmentation & Reassembly:** Breaks data into smaller segments at the sender and reassembles at the receiver
   2. **Port Addressing:** Uses **port numbers** to identify specific applications (e.g., 80 for HTTP)
   3. **Connection Control:** Provides **connection-oriented** (TCP) or **connectionless** (UDP) service
   4. **Flow Control:** Ensures the sender doesn't overwhelm the receiver

5. **Error Control:** Detects and retransmits lost/corrupted segments
6. **Multiplexing & Demultiplexing:** Allows multiple applications to share the network using different port numbers

Protocols Used in Transport Layer

1. **TCP (Transmisson Control Protocol):** it is connection-oriented protocol, Ensures reliable, ordered and error-free delivery.it is slower due to overhead.
2. **UDP (Used Datagram Protocol):** Connectionless, lightweight, Faster but unreliable no guarantees of delivery, ordering or error checking. Suitable for real time application, gives high speed.

Port Number in Transport Layer

Port-Type	Range	Use
Well-known	0-1023	Reserved for common protocols (HTTP, FTP, SSH)
Registered	1024-49151	For user-defined apps (e.g. MySQL, MongoDB)
Dynamic/Private	49152-65535	Used for temporary (ephemeral) connections

Connection Establishment in TCP (3-way Handshake)

1. **SYN** – Client sends SYN to initiate connection
2. **SYN-ACK** – Server acknowledges with SYN-ACK
3. **ACK** – Client replies with ACK → Connection established. Ensures both parties are ready to communicate before data is sent.

TCP Header Structure

1. **Source** Port: Sender's port number.
2. **Destination** Port: Receiver's port number
3. **Sequence** Number: Order of bytes in the message
4. **Acknowledgement** Number: Confirms receipt of data
5. **Flags**: SYN, ACK, FIN, etc.
6. **Window** Size: Flow Control
7. **Checksum**: Error checking

UDP Header Structure (Simple)

1. **Source Port:** Sender port
2. **Destination Port:** Receiver port
3. **Length**: Length of the data
4. **Checksum**: Error detection

5. **Network layer:** it works for communicating to other networks, this is where router is comes in the game, it has function of logical addressing or ip addressing, it assign ip address to network to ensure the data is received at the desired destination. Load balancing. Computer to computer. It is responsible for delivering **data Packets** from source host to destination host across multiple networks.

Functions of the Network Layer:

1. **Logical Addressing:** Assigns IP addresses to identify source and destination.
2. **Routing: Determines** the optimal path from source to destination.
3. Packet Forwarding: Moves packets from router to router towards destination.
4. **Fragmentation & Reassembly:** Breaks large packets into smaller chunks for transmission.
5. **Error Handling:** Reports errors like unreachable destination (ICMP)
6. **Traffic Control:** Manages data traffic to avoid congestion

Important Concepts:

1. **Logical Addressing (IP addressing)**
    - Unlike MAC addresses which are physical and fixed, IP addresses are logical and can change
    - Used to uniquely identify devices on a network.

    Types of IP

    - IVP4 (192.168.1.1): 32-bit
    - IPV6 (2001:0db8:85a3::8a2e:0370:7334) – 128 bit.

    Difference Between IPV4 and IPV6

Features	IPv4	IPv6
Address Length	32 Bits	128 Bits
Address Format	Decimal (192.168.0.1)	Hexadecimal (2001:db8::1)
Total address	~4.3 billion (2^32)	~340 undecillion (2^128) almost Unlimited
Header Size	20 bytes	40 bytes
Header Complexity	Complex	Simplifed
Address configuration	Manual to DHCP	Auto (Stateless) or DHCPv6
Broadcast	Supports broadcasting	No broadcasting (multicast)
NAT Support	Required due to limited IPs	Not needed (enough IPs)
Fragmentation	Done by sender and routers	Only Sender does fragmentation
Checksum	Present in header	Removed to improve performance
Mobility support	Limited	Better support for mobile devices

2. **Routing:** They decide the best path based on: Destination IP address, Routing tables, Routing protocols (e.g. OSPF, BGP, RIP).
   Types of routing:
   a. **Static routing:** Manually configured by the next administrator. Routes do not change automatically if the network changes. low flexible, suitable for small and simple networks.
   b. **Dynamic Routing:** Routers automatically learn and update routes using routing protocols. Can detect network changes and re-route traffic. High flexibility. Suitable for Large, scalable neworks.

   Key Terms in Routing

   a. **Routing table:** List of routes to reach different networks.
   b. **Hop Count:** Number of routers between source and destination.
   c. **Next Hop:** The next router a packet should go to.
   d. **Metric:** Value used to choose best route
   e. **Convergence:** Time taken for routers to update and agree routes.

3. **Packet switching:** The data is divided into packets, Each packet may take different routes to reach the destination. Once all packets arrive, they are reassembled.
4. **Fragmentation and Reassembly:** If a packet is too large for a particular network (Like Ethernet). its broken (fragmented). The destination device reassembles it. Controlled by MTU.

Protocols working at Network Layer:

1. **IP (Internet Protocol):** Provides logical addressing and routing (IPV4, IPV6).
2. **ICMP (Internet Control Message Protocol): Reports** errors and diagnostics (Used by **ping**).
3. **ARP (Address Resolution Protocol):** Maps IP addresses to MAC addresses.
4. **RARP (Reverse ARP):** Maps MAC address to IP address.
5. **IPSec:** Provides security at the IP layer.
6. **OSPF/ RIP/ BGP:** Routing protocols for finding best paths.

6. **Data-Link layer:** data link will receive the data from network layer and it allows to talk directly with your computer. The physical addressing done by the data-link layer it known for MAC addressing to form a frame. mac address is a 12 digit number, it also control how the data is placed in the computer.

   Functions of Data Link Layer:

   1. **Framing:** Groups bits into logical units called frames.
   2. **Physical Addressing:** Adds MAC addresses to identify sender and receiver. it is a hardware address unique to each device. it is 48 bits address.
   3. **Error Detection and correction:** Uses checksums or CRC to detect/correct transmission errors.
   4. **Flow Control:** Prevents fast sender from overwhelming slower receiver.
   5. **Access Control (MAC):** Controls access to the shared medium.
   6. **Acknowledgment:** Confirms successfully receipt of frames

   Devices working at Data Link Layer:

   1. **Switch**: Forwards frames based on MAC address.
   2. **Bridge**: connects two LANs and filters traffic using MAC
   3. **Network Interface Card (NIC):** Uses MAC address and handles frame-level operations

7. **Physical layer:** it handles all the hardware how data is transfer from wires and cables. it Is responsible for the transmission of raw bits (0s and 1s) over a physical medium such as cables, radio waves, or fiber optics.

   Functions of Physical layer:

1. **Bit Transmission:** Converts frames (from data link layer) into signals.
2. **Media type Definition:** defines the physical medium (cable, fiber, air).
3. **Signal encoding:** Uses encoding schemes to represent bits.
4. **Data Rate Control:** Defines bit rate – how fast bits are sent (bps).
5. **Synchronization:** Ensures sender and receiver are in sync to interpret bit boundaries.
6. **Topology & physical Design:** Defines how devices are physically connected (star, bus, ring).
7. **Transmission Mode:** Supports simplex, half-duplex, and full-duplex communication. (one-way, two-way but one at a time, Two-way)
8. **Interface Specifications:** Defines pins, voltages, connectors

Data Flow Example (Top to bottom OSI)

1. **Application Layer:** User sends a message.
2. **Transport Layer:** It becomes a segment.
3. **Network Layer:** It goes an IP packet.
4. **Data Link Layer: It** gets a frame (adds MAC).
5. **Physical Layer:** converts it into electrical/light signals and sends over cable/wireless.

There is one more another model present

TCP/IP Model: it is also called Internet Protocol Suite is a **real-world, practical framework** used to design and implement network communication protocols. It forms the **foundation of the internet.** It is 4-layer model that defines how data should be packaged, addressed, transmitted, routed and a received over a network.

Layer No.	TCP/IP Layer	OSI Equivalent
4	Application	Application + Presentation + Session
3	Transport	Transport
2	Internet	Network
1	Network Access	Data Link + Physical

1. **Application Layer:** It is Closest to the user, Includes high-level protocols used for communication. Examples: HTTP, FTP, SMTP, DNS, DHCP,SNMP.
   Functions: Data formatting, Encryption, Dialog control, User authentication.
2. **Transport Layer:** Ensures reliable data transmission, Breaks data into segments, Uses ports to identify applications.
   Protocols:
   - **TCP (Transmission Control Protocol):** Reliable, connection-oriented.
   - **UDP(User Datagram Protocol):** Fast, Connectionless, unreliable.
3. **Internet Layer:** Responsible for logical addressing and routing, Delivers packets across networks.
   Protocol:
   - **IP(Internet Protocol):** Provides logical addressing (IPV4/IPV6)
   - **ICMP:** error messages and diagnostics (e.g. ping)
   - **ARP:** resolves IP to MAC
   - **RARP:** MAC to IP
4. **Network Access Layer (Link Layer):** Handles the physical transmission of data, Includes framing, MAC addressing, error detection. Example: Ethernet, Wi-Fi, PPP, DSL, MAC.
   Functions: Converts data into bits, Determines how bits are transmitted (electrical, radio, optical)

Some extra information:

1. **Repeaters:** A **repeater** is a **network device** that **receives a signal**, **amplifies it (or regenerates it)**, and then **retransmits it** to extend the range of a network. A **repeater** helps in **boosting weak or corrupted signals** over long distances in a network to maintain data integrity. Works at the physical layer of OSI model. Used to extend the range of LANs or other networks.
2. **Hub:** A **hub** is a **basic, non-intelligent device** used to connect multiple devices in a network. It **operates at Layer 1** of the OSI model. It **broadcasts data to all ports**, creating **a single collision domain**. Mostly **replaced by switches** today due to performance and security limitations.

3. **Bridge:** A **bridge** connects two or more LAN segments and **filters traffic using MAC addresses**. It works at the **Data Link Layer (Layer 2)** of the OSI model. Helps in **reducing traffic, avoiding collisions**, and **improving network performance**. Bridges are mostly **replaced by switches** in modern networks, but the concept remains essential in networking.
4. **Switch:** A **switch** is a smart network device that connects devices within a LAN and **forwards data based on MAC addresses**. Operates at the **Data Link Layer (Layer 2)** of the OSI model. **Reduces collisions, increases speed, and improves efficiency**. Commonly used in modern LANs and **replaces hubs and bridges**.
5. **Routers:** A **router** is a smart network device that connects **different networks** and **routes packets** using **IP addresses**. Operates at the **Network Layer (Layer 3)** of the OSI model Essential for **internet connectivity**, **NAT**, and **path selection**. Most home routers are **Wi-Fi routers** with built-in **switch and firewall** functions.
6. **Gateway:** A **gateway** is a **protocol converter** that enables communication between **two different networks** (e.g., TCP/IP ↔ Bluetooth). Works at multiple layers of the OSI model (mainly **Layer 3 and above**). Acts as a **translator**, making it more advanced than routers or switches. **Routers route**, but **gateways translate** and connect **unlike systems**.
7. **Brouter:** A **brouter** is a **hybrid device** that performs both **routing (Layer 3)** and **bridging (Layer 2)**. It is used in networks where **both routable and non-routable protocols** exist. Brouters are **less common today** as most modern networks use only **routable protocols like IP**, and advanced routers handle all tasks.
8. **Sockets:** A **socket** is the **interface between the application and transport layers** in the OSI model. It enables **network communication** between client and server applications. It is defined by an **IP + Port**. TCP sockets ensure **reliable, ordered** communication.UDP sockets are **faster but not reliable** (no guarantee of delivery).
9. **PORTS:** A **port** is a **logical endpoint** in a computer network used to **identify specific processes or services** on a device. It helps in **directing the incoming or outgoing network data** to the correct application. Ports work alongside **IP addresses** to complete network communication. There are **65,535 ports**, divided into **well-known**, **registered**, and **dynamic** ranges. Common ports: **80 (HTTP)**, **443 (HTTPS)**, **22 (SSH)**, **53 (DNS)**.

Port number Range: 0-1023 (Standard/common services (HTTP.FTP,etc)), 1024-49151(Registered ports for user-defined applicaion), 49152-65535(Dynamic/private ports fro temporary use of client-side communication)

Server Architecture:

Server architecture refers to the **design and structure** of how a **server system is built**, how it **operates**, and how it **communicates** with clients or other systems. it defines: how data is processed and stored, How resources are managed, How requests from clients are handled.

Types of Server Architecture

1. **Client-Server Architecture (Traditional):** One or more clients (like browsers) interact with a central server. Server processes requests and returns responses.
2. **Three-Tier Architecture (Most common web architecutre): Presentation Layer (Client):** User interface (browser/app) **Application Layer (Server):** Business logic (Node.js, Java, Python) **Database Layer:** Stores and retrieves data (MySQL, MongoDB)
3. **Microservices Architecture:** Application is split into small, independent services, Each service handles a specific function (auth, payment, etc.)
4. **Monolithic Architecture:** Entire application runs as a single unit.

HTTP & HTTPS:

HTTP (HyperText Transfer Protocol) is an **application layer protocol** used for **transferring data over the web**.
It defines **how web clients (like browsers)** and **web servers communicate** to exchange information such as **HTML pages, images, videos, and data APIs**. it is stateless, connectionless and text-based.

Structure of an HTTP Request:

- **Request Line:** Method + URL + HTTP version (e.g., GET /index.html HTTP/1.1).
- **Headers**: Metadata like Host, User-Agent, Accept

- **Body**: (Optional) – sent with methods like POST or PUT

Structure of an HTTP Response:

- **Status Line:** HTTP version + status code + message (e.g., HTTP/1.1 200 OK)
- **Headers:** Metadata like Content-Type, Content-Length.
- **Body:** Actual content like HTML, JSON, images, etc.

Common HTTP Methods:

- **GET**: Request data from server (read-only)
- **POST**: Send data to server (Create new resource)
- **PUT**: Update existing resource
- **DELETE**: Remove resource
- **PATCH**: Partially update resource
- **HEAD**: Like GET, but returns only headers
- **OPTIONS**: Returns allowed methods

Common HTTP Status Code:

- **200**: ok – successful request
- **201**: Created – New resource created
- **400**: Bad request – Invalid input
- **401**: Unauthorized Authentication required
- **403**: Forbidden Access Denied
- **404**: Not found Resource not found
- **500**: Internal Server Error Server-side issue

How Email works?

1. **Key players & Terminology**

Abbreviation	Full Name	Role
MUA	Mail User Agent (e.g. Gmail app, Outlook)	Where the user composes/reads mail
MTA	Mail Transfer Agent (e.g. Postfix, Exim)	Post-office truck: relays mail between servers via SMTP
MDA	Mail Delievery Agent(e.g. Dovecot, procmail)	Final sorter: drops mail into the recipient's mailbox.
SMTP	Simple mail transfer rotocol	Protocol MTAs & MUAs use to send mail

POP3/IMAP	Post office Protocol / Internet message access protocol	Protocol MUAs use to retrieve mail (POP3 = download & delete;IMAP = sync)
DNS MX Record	Mail eXchanger record	Tells the world which server accepts mail for a domain

2. **Anatomy of an E-mail**

   Header section

   From: aadii@example.com

   To: friend@domain.com

   Subject: Hello

   ...other headers (Date, Message-ID, MIME-Version, etc.)

   Blank line

   Body (plain text or HTML, plus MIME parts for attachments)

3. **Life of an E-mail (Step-by-step)**

   a. **Compose – MUA -> Outgoing SMTP**

- You press **Send** in Gmail (MUA).
- MUA opens a TCP connection (usually 587 or 465 with TLS) to your domain's **outbound SMTP server** (first MTA).
- Authentication happens (SMTP AUTH) to prevent open-relay abuse.

   b. **Local Relay & Queue – MTA:** The MTA builds an SMTP envelope (MAIL FROM, RCPT TO, etc.) and sticks the message in its **queue**.

   c. **DNS Lookup – Finding the Destination**

      - MTA queries DNS for **MX records** of domain.com.
      - Suppose DNS returns 10 mx1.domain.com, 20 mx2.domain.com.

   d. **Transfer – MTA #1 -> MTA #2 (Destination)**

      - MTA establishes an SMTP session with mx1.domain.com on port 25 (or 25 over STARTTLS).
      - If mx1 is down, it tries mx2.

   e. **Reception & Local Delivery — MTA #2 → MDA**

      - Destination MTA checks spam rules, SPF, DKIM, DMARC, virus scanners.
      - If accepted, it hands the message to an **MDA** or drops it directly into the user's **mailbox file/folder** (/var/mail/friend, or in IMAP's Maildir).

   f. **Retrieval – MUA <- POP3/IMAP**

      - The recipient's MUA connects (usually port 993 for IMAPS, 995 for POP3S).

- IMAP leaves mail on the server and syncs flags; POP3 typically downloads then deletes (configurable).
- Message shows up in the inbox.

4. **Security & Trust Chain**

Technology	**What it Adds**
TLS (STARTTLS/SMTPS)	Encrypts SMTP hop so contents & credentials aren't sniffed.
SPF(Sender Policy Framework)	Publishes who may send mail for a domain (DNS TXT)
DKM (DomainKeys Identified Mail)	MTA signs mail with a private key; recipient verifies via DNS-published public key
DMARC	Policy telling recipients how to react when SPF/DKIM fail
S/MIME/PGP	End-to-end encryption & signing (optional, user-level)

5. **Important Ports & status Codes**
    - 25 – SMTP MTA-to-MTA (plus STARTTLS)
    - 465 – SMTPS (legacy, implicit TLS)
    - 587 – Submission port (SMTP + STARTTLS)
    - 110/995 POP#/POP3S
    - 143/993 – IMAP/IMAPS

DNS (Domain name system)?

DNS (Domain Name System) is a system that **translates human-readable domain names** like www.google.com into **IP addresses** like 142.250.195.68, which computers use to communicate with each other over the internet.

How DNS works?

When you type www.example.com into your browser:

1. **Browser Cache Check:** The browser checks if it already knows the IP address (from cache).
2. **OS Cache Check:** If not found, the Operating System checks its local DNS cache.
3. **Query Sent to Recursive DNS Resolver:** If still unresolved, a query is sent to a **recursive DNS server** (usually provided by your ISP or Google DNS: 8.8.8.8).
4. **Recursive Resolver Checks**:

a. **Root DNS Server:** Tells the resolver which **Top-Level Domain (TLD)** server to contact (e.g., .com, .org).
b. **TLD DNS Server:** Directs the resolver to the **Authoritative Name Server** for the domain.
c. **Authoritative Name Server:** Contains the actual IP address for www.example.com and returns it.

5. **Response to Client:** The resolver returns the IP to your browser, which then contacts the web server to fetch the website.

Types of DNS Servers:

- **Recursive Resolver**: Finds the IP on behalf of the client
- **Root Server:** Directs to TLD server
- **TLD Server:** Directs to authoritative server
- **Authoritative Server:** Has the final answer (IP address of domain)